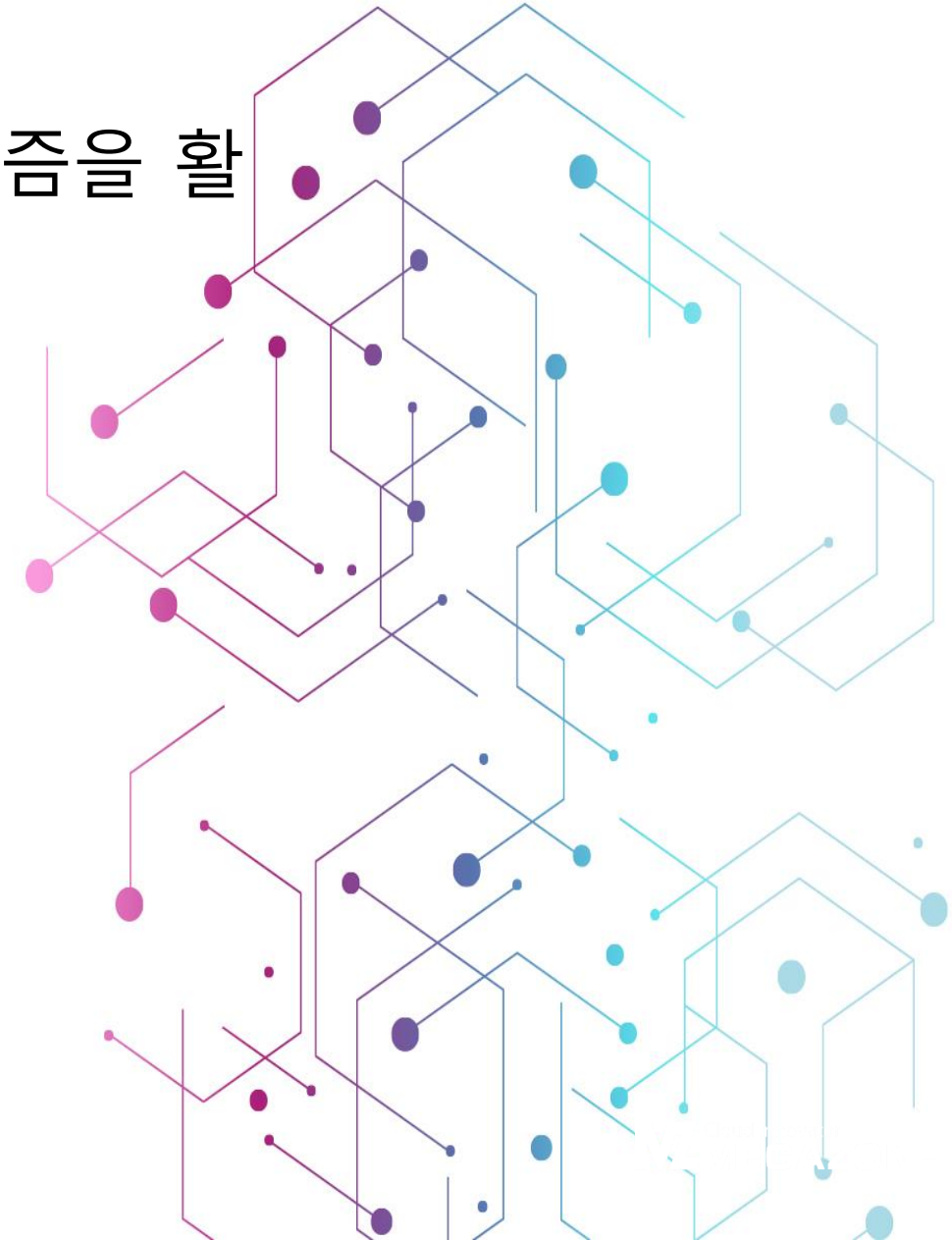
한국R사용자커뮤니티(R Korea)

EM 기반의 SoftImpute 알고리즘을 활용한 추천알고리즘 적용 사례



MEGAZONE CLOUD
Data Service Center

이용혁

2019.10.25

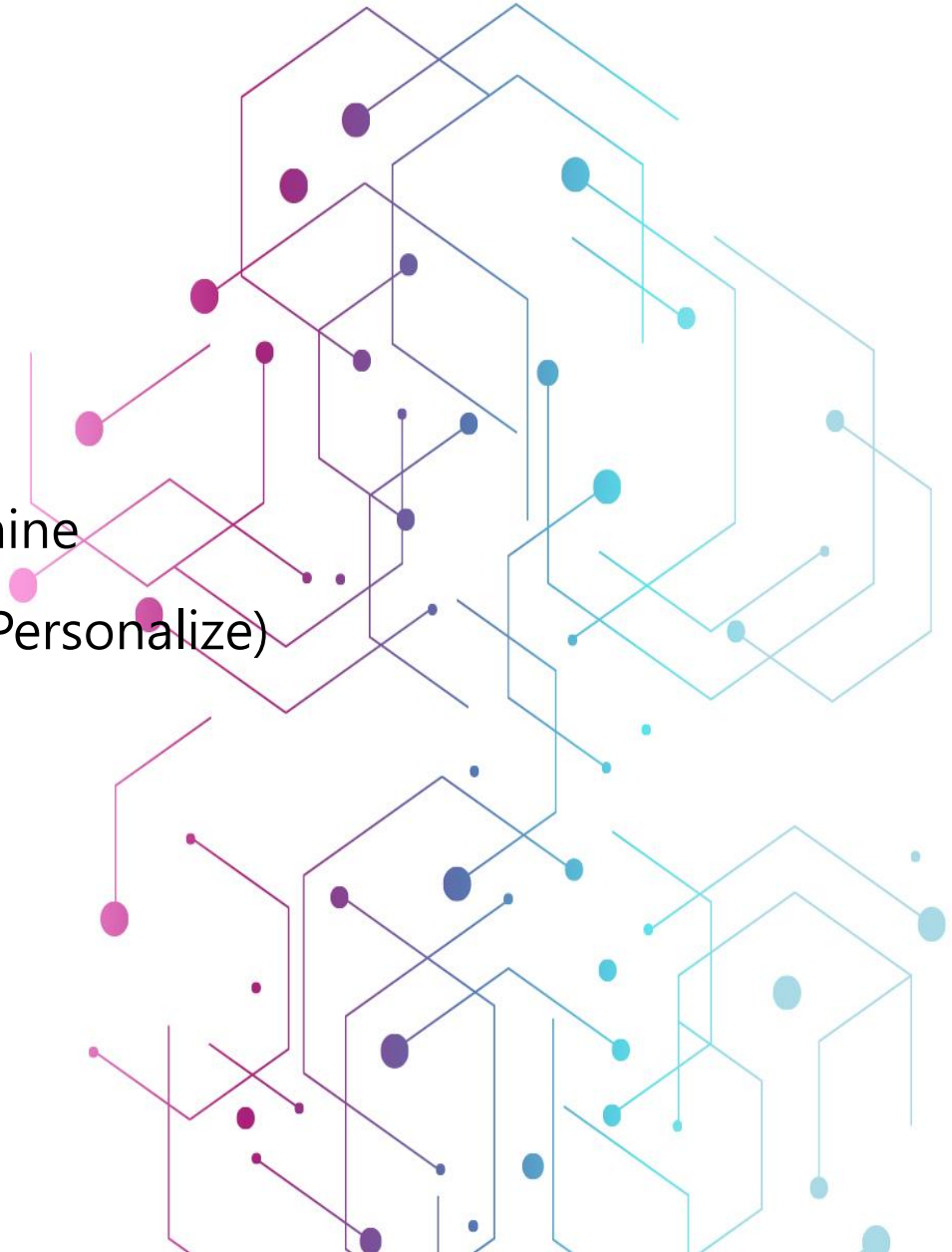
Summary

SoftImpute

Factorization Machine

DeepFM(Amazon Personalize)

목차



Matrix Completion

Matrix Completion문제는 추천 영역에서 우수한 성능을 보임이 알려진 이후 이론적으로 발전해왔으며 넷플릭스 경진대회를 기점으로 대중적으로도 알려지게 됨

Problem setting

$$M_0 = \left(M_{0,ij}\right)_{i=1,\dots,n,j=1,\dots,d}$$

관찰된 값

$$\mathbf{M} = \left(M_{0,ij}\right)_{i=1,\dots,n,j=1,\dots,d}$$

$$with \;\; M_{ij} = y_{ij}\left(M_{0,ij} + \varepsilon_{ij}\right)$$

$$where\; y_{ij} = 1\; if\; observed$$

특징

- Low-rank matrix
- Not fully observed
- Sparse data
- Error contaminated

Matrix Completion

Low-rank 행렬을 복구하는 것은 NP-hard problem 이지만 convex relaxation을 적용하면 기본적으로 행렬 분해 방법(SVD)을 활용할 수 있게 됨

Algorithm

1. Initial value $\mathbf{Z}_{old} = 0$.
2. Calculate $\mathbf{A} = P_\Omega(\mathbf{M}) + P_\Omega^{\perp}(\mathbf{Z}_{old})$ which admits the following singular value decomposition

$$A = \sum_{i=0}^{\min(n,d)} d_i U_i V_i^{\mathsf{T}}$$

where d_i is the i th largest singular value for $\mathbf{A}$ and $\mathbf{U}_i$ and $\mathbf{V}_i$ are the corresponding left and right singular vectors, respectively.

3. Calculate $\mathbf{Z}_{new} = \sum_{i=1}^{\min(n,d)} (d_i - \lambda) + U_i V_i^{\mathsf{T}}$
4.
   a. If $\frac{\|\mathbf{Z}old - \mathbf{Z}new\|_F^2}{\|\mathbf{Z}old\|_F^2} \leq \varepsilon$, then exit.
   b. Otherwise, $\mathbf{Z}old = \mathbf{Z}new$. Then go to step 1.

결과

- SoftImpute 알고리즘을 사용하여 converge함을 보임
- 단, 성능은 tuning parameter λ 에 영향을 받음

$$\min_Z \frac{1}{2} \|P_\Omega(\mathbf{M}) - P_\Omega(\mathbf{Z})\|_F^2 + \lambda \|\mathbf{Z}\|_*$$

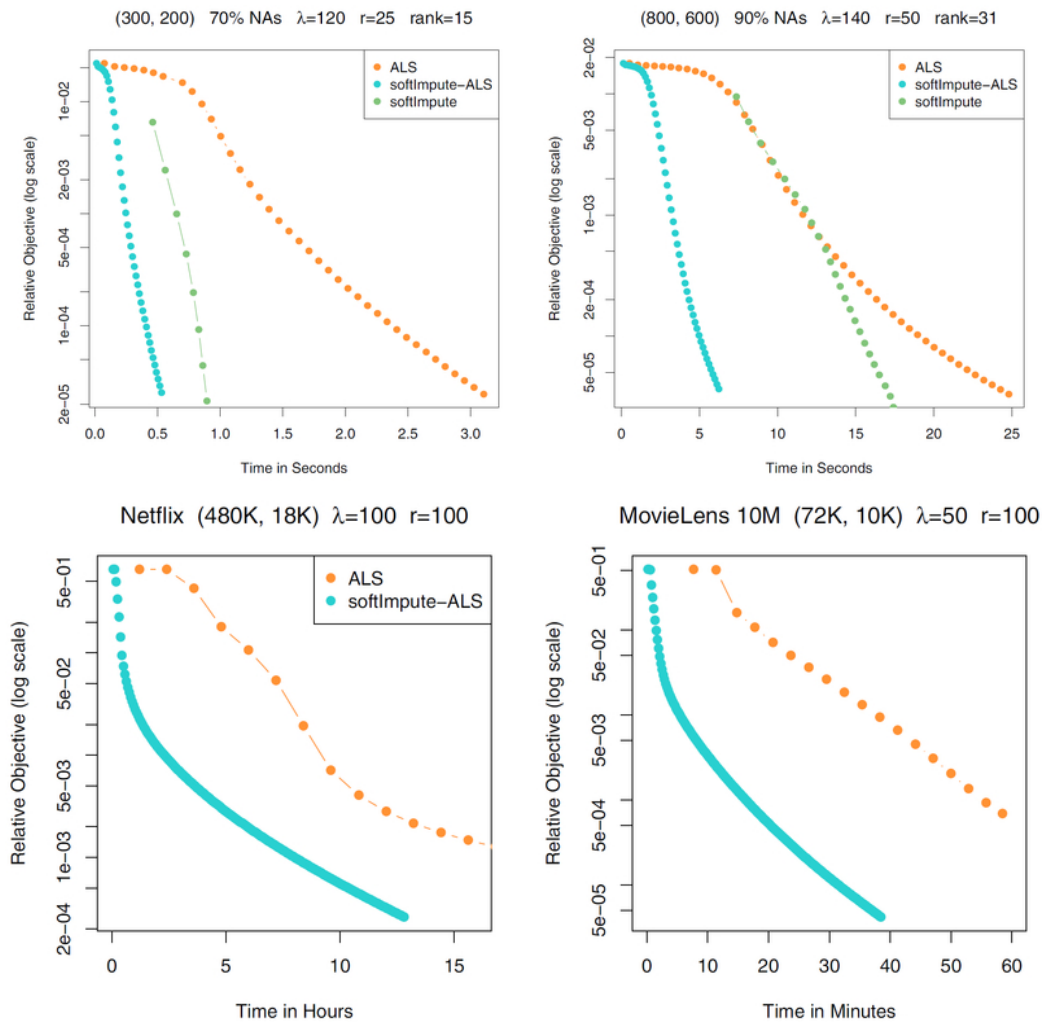
- Convex relaxation

SoftImpute

softImpute 에서는 SVD 및 ALS 두 가지 타입의 알고리즘을 제공하고 있으며 속도 측면에서보면 ALS가 더 빠르게 수렴함

Singular Value Decomposition(SVD)	Alternating Least Squares(ALS)
반복적 SVD 계산을 활용하여 Matrix Completion 수행 Mazumder et al (2010)	Alternating Ridge regression at each stage filling in the missing entries with the latest estimates.
$d [1] 4.752691 2.056353	$d [1] 4.853467 2.041956
29 : obj 0.0077 ratio 9.168035e-06	24 : obj 0.00774 ratio 8.803524e-06

SoftImpute - ALS



SoftImpute - ALS

SoftImpute-ALS 알고리즘은 upper bound를 활용한다는 측면에서 EM기반(혹은 더 일반적으로 MM-style)* 알고리즘으로 볼 수 있음

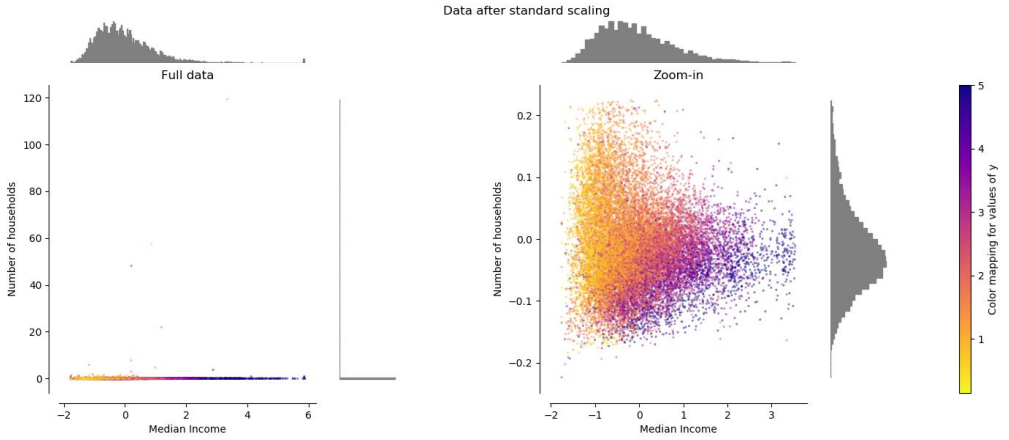
$$\underset{A,B}{\text{minimize}} \quad F(A,B) := \frac{1}{2}\|P_\Omega\left(X - AB^T\right)\|_F^2 + \frac{\lambda}{2}\left(\|A\|_F^2 + \|B\|_F^2\right),$$

$$\begin{aligned} \widehat{A} &= U_r \mathcal{S}_\lambda(D_r)^{\frac{1}{2}} \\ \widehat{B} &= V_r \mathcal{S}_\lambda(D_r)^{\frac{1}{2}}, \end{aligned}$$

* Missing value가 존재 할 때 MLE 풀기 위해 적용할 수 있는 알고리즘

Movielens Exercise

Movielens-100k 데이터를 활용하는 추천알고리즘으로 SoftImpute를 수행하기 위해서는 여러번의 tunning 작업을 수행해볼 것을 권장



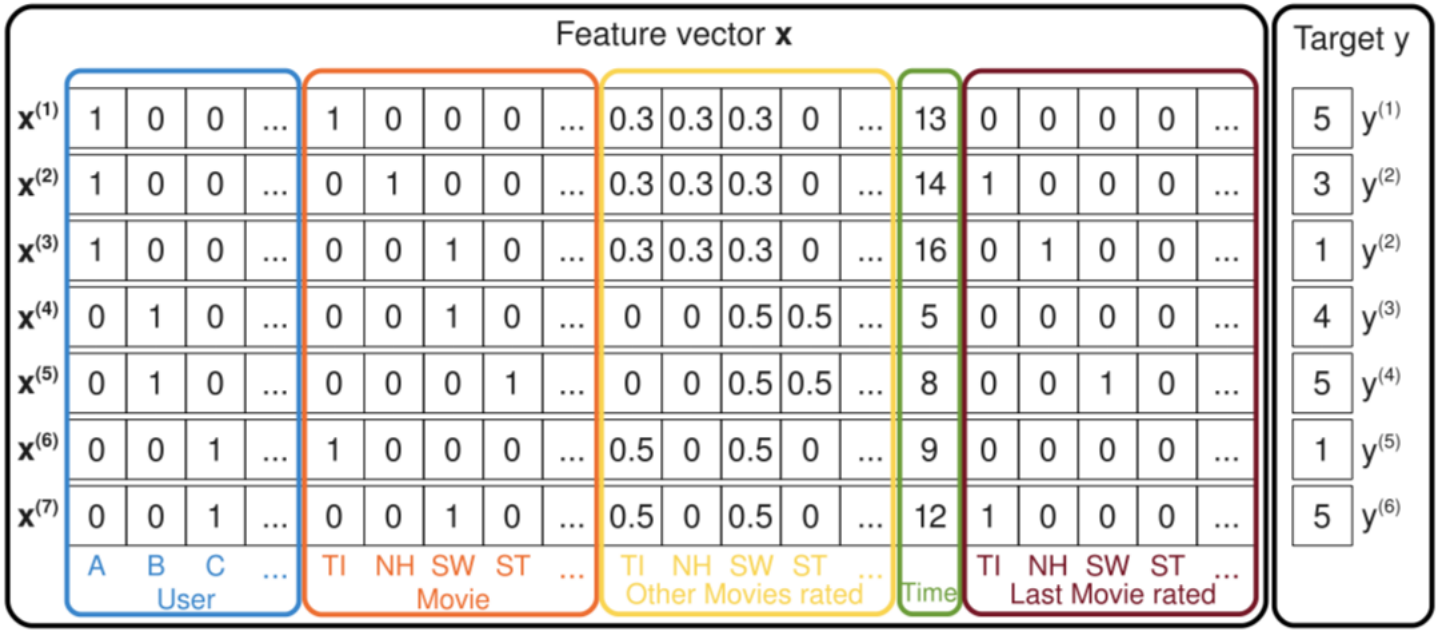
$$\min_{z} \frac{1}{2} \| P_{\Omega}(\mathbf{M}) - P_{\Omega}(\mathbf{Z}) \|_F^2 + \lambda \| \mathbf{Z} \|_*$$

- Convex relaxation

SoftImpute::biScale

Factorization Machine

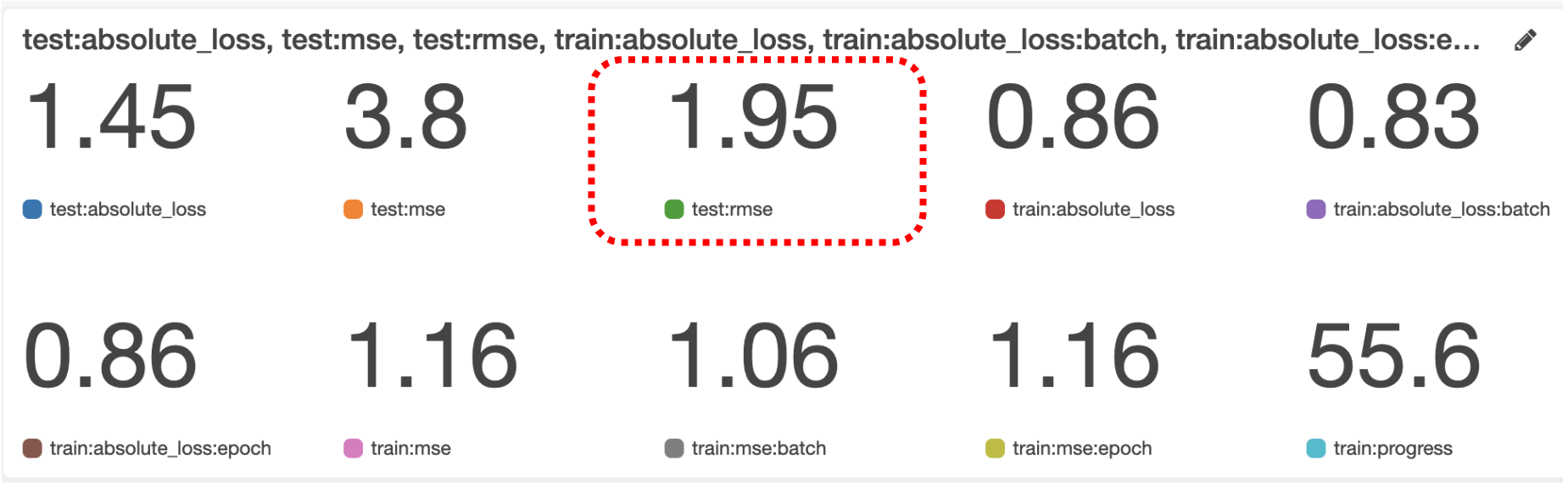
회기 문제 와 이진 분류 문제 두 가지 모드를 모두 지원하므로 영화 평점 이외에 이력 정보 및 변수를 활용하여 추천하고자 하는 상품을 예측할 수 있음



$$\hat{y} = w_0 + \sum_i w_i x_i + \sum_i \sum_{j>i} < v_i, v_j > x_i x_j$$

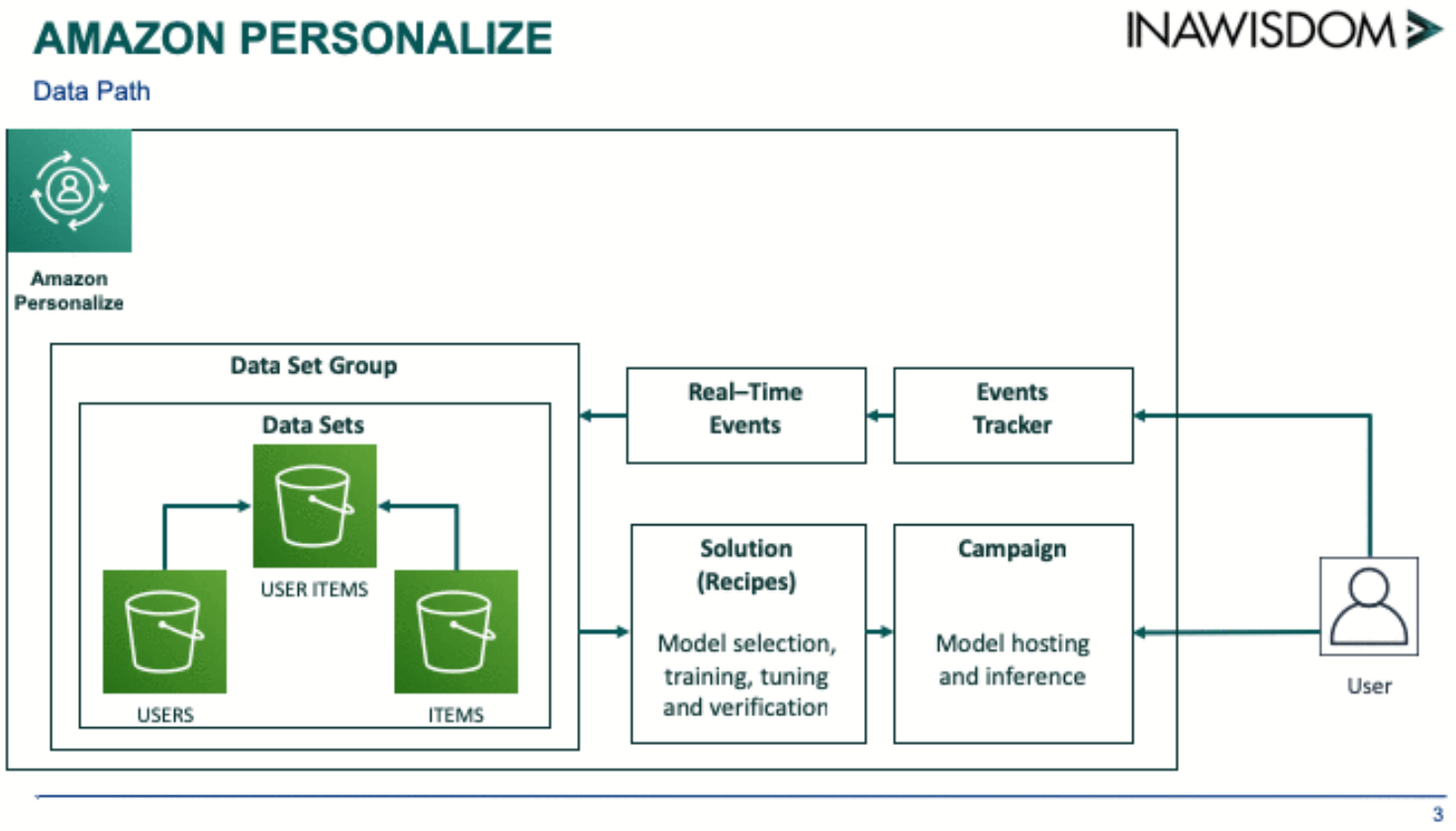
Movielens Example

단순히 userId 와 movieId만을 가지고 Factorizaiton Machine을 학습시킨 결과



[별첨] Amazon Personalize

메타데이터를 고려할 수 있는 데이터 그룹을 생성하고 실시간으로 API호출을 요청할 수 있음



THANK YOU